2000

Use what you've got: Steps toward opportunistic computing

David R. McGee and Philip R. Cohen
Center for Human Computer Communication
Department of Computer Science and Engineering
Oregon Graduate Institute
Portland, OR 97006 USA
+1 503 748 1602
{dmcgee, pcohen}@cse.ogi.edu
<http://www.cse.ogi.edu/CHCC/>

ABSTRACT

We define *opportunistic computing environments* as augmented reality environments that attempt to minimally perturb existing work practices and allow users to opportunistically take advantage of them. In this paper, we present Rasa, an opportunistic computing environment for military command and control that augments the physical objects on a command post map by observing and understanding the users' speech, pen, and touch-based multimodal language. We give a thorough account of Rasa's underlying multiagent framework, recognition, understanding, and multimodal integration components. Finally, we examine three properties of language—generativity, comprehensibility, and compositionality—that render it suitable as an augmentation scheme, and we compare these properties to those of current tagging technologies and approaches.

Keywords

augmented reality, mixed reality, multimodal interfaces, tangible interfaces, and invisible interfaces

INTRODUCTION

This paper describes an *opportunistic computing* environment—an augmented reality environment that attempts to minimally perturb existing work practices and allows users to opportunistically take advantage of existing tools and processes.

We begin by describing a common work practice we observed in a military command post, where we found people already augmenting physical objects. Based on these observations, we developed a set of design constraints to support the development of an augmented reality environment that is able to employ these preexisting augmentations. In a recent paper, we showed that contemporary object tagging approaches fail to meet these augmentation criteria [18]. Here we discuss why language, due to its generativity, compositionality and comprehensibility meets each criterion, and why that makes it an especially attractive augmentation method. Finally, we present a detailed description of Rasa—a system that understands the language officers use in military command posts and updates digital command and control systems by capturing, recognizing, and understanding this language used to augment physical objects on a paper map.

WORK PRACTICE

At Ft. Leavenworth, Kansas and at other military bases, we observed commanders and their subordinates engaging in command and control of armed forces. The photograph in Figure 1 was taken during an especially frenetic period in the command post.

On the left is a rear-projected SmartBoard™ and on the right is a Sun Microsystems workstation. Several other

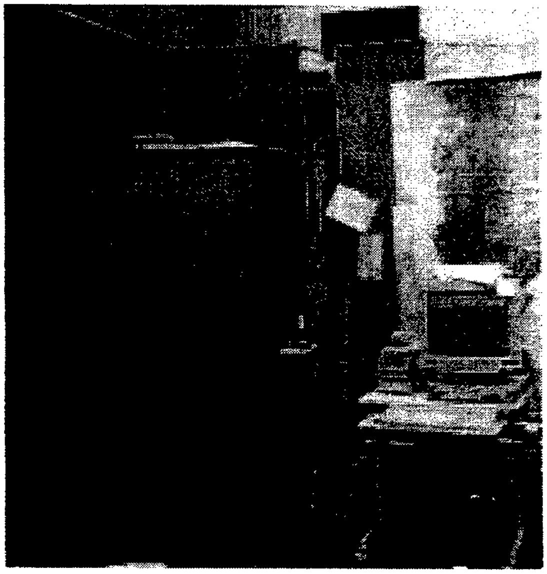

Figure 1. State of the art military command and control systems in action. Photo courtesy of William Scherlis.

systems are in the immediate foreground. On each is one of the Army's latest command and control software systems. Notice that no one is using these systems.

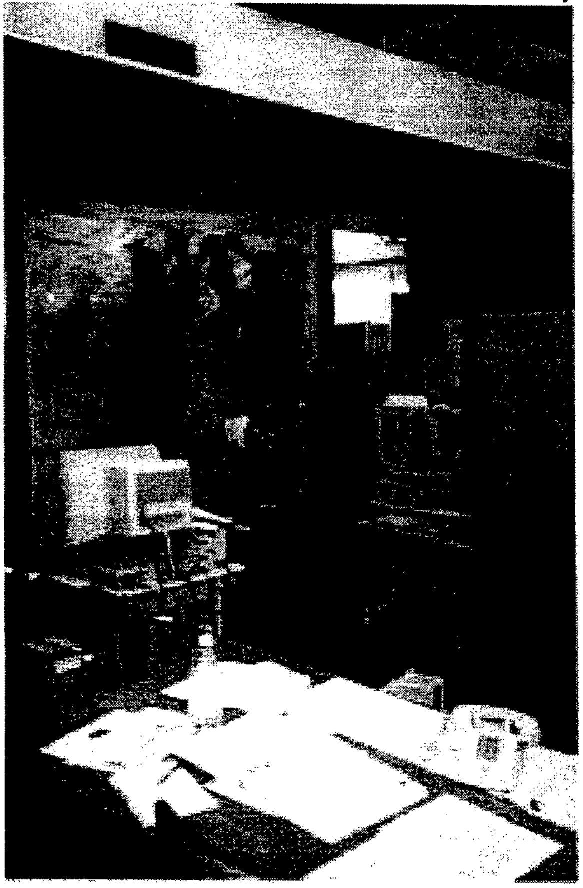

Figure 2. What commanders prefer. Photo courtesy of William Scherlis

During this critical phase of work, the commander and his staff have chosen to use a different set of tools than the ones designed for this task. They have quite purposefully turned their backs on computer-based tools and graphical user interfaces. Rather, we see that they have chosen to use a large 8-foot high by 4-foot wide paper map, arrayed with Post-it™ notes (Figure 2).

This large map, one of the primary command and control tools used by commanders, has a two-fold purpose: 1) to depict the terrain, its occupants (military units consisting of soldiers and machinery), their position, and capabilities; and 2) to overlay that information with a graphical rendition of the daily plan for the force. The map is kept up-to-date by constant communication both up and down their organizational hierarchy.

The overlays are Mylar sheets reproduced each night by planners. Copies are shared amongst relevant units. A symbol representing a unit's functional composition and size is sketched in ink on each Post-it. As unit positions arrive over the radio, the Post-its representing these units are moved to their respective positions on the map.

The users establish associations between the Post-it notes and their real-world entities with a standardized language used since Napoleon's time that is capable of denoting thousands of units types.

The officers choose paper for a variety of reasons. It has extremely high resolution. Moreover, it is malleable, cheap, and lightweight, and thus can be rolled up and taken anywhere. Additionally, people like to handle physical objects as they collaborate. As officers debate the reliability of sensor reports and human observation to determine the actual position of units in the field, they jab at locations on the map where conflicts may arise. They also pick up Post-it notes and hold them in their hand while they debate a course of action.

Rasa perceives the augmentations resulting from the users' interacting multimodally with these physical objects by understanding this language. Consequently users can continue to employ familiar tools and procedures, which in turn create automatic couplings to the digital world. In the next section, we examine other methods that have been used to augment environments.

Augmenting physical objects

Researchers have used two methods to augment physical objects:

- sensing physical properties (weight [24], shape, color, size, location)
- affixing sensable properties, hereafter called simply "tags" (bar codes [9], glyphs [19, 25], radio-frequency identifier tags [26], etc.)

Most researchers will agree that sensing the physical properties of arbitrary objects to identify them uniquely is still quite difficult, except in tightly constrained environments. The Perceptive Workbench [15] is one notable example. Thus, most unencumbering augmented reality systems rely on tags to augment objects.

We considered each of these tagging approaches when we began to design Rasa, but none of them met the physical constraints needed to augment the Post-it notes. Moreover, none fulfilled the design constraints discussed below and in detail in [18].

Minimality Constraint	Changes to the work practice must be minimal. The system should work with user's current tools, language, and conventions.
Human Performance Constraint	Multiple end-users must be able to perform augmentations.
Malleability Constraint	Because users gain information about the real world object over time, the meaning of an augmentation should

	be changeable; at a minimum, it should be incrementally so.
Human Understanding Constraint	The users must be able to perceive and understand their own augmentations unaided by technology. Moreover, multiple users should be able to do likewise, even if neither are spatially nor temporally co-present. Users must also understand what the augmentation entails about the corresponding objects in the real world.
Robustness Constraint	The work must be able to continue without interruption should the system fail.

None of the prior tagging methods provides a persistent representation of the augmentation that would make it robust to every kind of system failure (communications, power, etc.), a common occurrence in this environment. Users have no way of deciphering bar codes and glyphs without computational aid. If the tag readers and other systems fail, people can no longer understand what the tag means. With none of these methods is there a natural way for the end-user to create the augmentations. Finally, none of these tagging methods could be introduced into the work practice without engendering a great degree of change.

In the next section, we discuss how language overcomes these deficiencies. Moreover, we examine the properties of language that argue for its consideration as a tool for augmenting physical objects.

LANGUAGE

Language is generative, compositional, and comprehensible, and depending on whether language is written or spoken, it can be permanent or transitory, respectively. These attributes make language a particularly suitable candidate for creating augmentations. By "language," we mean an arrangement of perceptible "tokens" that have both structure and meaning. This definition is meant to subsume both natural spoken and written languages, as well as diagrammatic languages such as military symbology.

The military symbology language taught to all soldiers consists of shapes to indicate friend (rectangle) or foe (diamond), a set of lines, dots, or "X's" to indicate unit size (squad to army), and a large variety of symbols to indicate unit function (e.g., mechanized, air defense, etc.) denoted by combinations of meaningful diagrammatics. For example, an armored reconnaissance unit's symbol is a combination of the marks used for armor and for reconnaissance, as shown in Figure 3. In addition, the unit's reporting structure (e.g., First platoon, Charlie company) is often indicated by an abbreviation (1/C) written to the side of the symbol (see Figure 4). In virtue of these components and its compositional nature, the symbol language can denote thousands of units as well as their position in the unit hierarchy. Because the language is compositional, soldiers are able to understand and generate complex concepts in terms of their parts. Moreover, soldiers use the language to communicate the situation to others by arraying such symbols on written notes or other devices (e.g. pushpins) on the map.

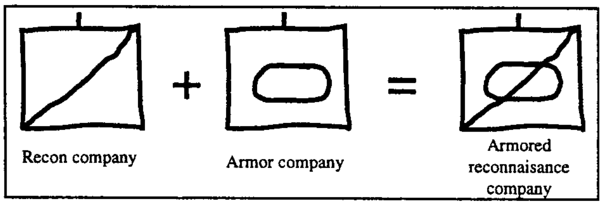

Figure 3. Composition of unit symbols

Since written language is permanent, it leaves behind a persistent trail as paper-based augmentations are incrementally applied. Not only can we understand shared languages unaided, we can often recognize the author of an utterance. Neither of these abilities is resident in tags.

Spoken language is convenient when the user would prefer a lack of persistence. For example, rather than update the permanent shared understanding conveyed by the map, spoken language is often used to name or refer to objects. This generativity is another unique property of language, which enables users to create references, placeholders, metaphors, symbols, etc. Users often name entities (e.g., "advanced guard") while writing their symbology on the Post-it notes, thereby making the notes a placeholder for entities in the real world.

None of these characteristic properties of language—generativity, compositionality, and comprehensibility—is present in tagging approaches to augmented reality systems. One might imagine designing a user interface to accompany the creation and reading of tags that offered these capabilities. However, any change to these highly learned behaviors, such as the introduction of computer interfaces, which is not minimally disruptive to the process is likely to be resisted.

In summary, the language used in the command post offers an ideal means for augmenting physical objects such as Post-it notes. However, in order to take advantage of the users' reliance on language, a system must be capable of understanding it. In the following section, we present the architecture for Rasa—a system that enables multimodal understanding of spoken and gestural language in such augmented environments, which is derived from our earlier work on QuickSet [7].

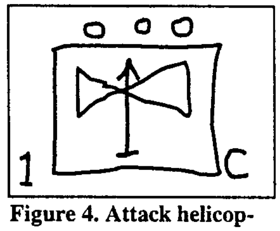

Figure 4. Attack helicopter symbol called First platoon, Charlie company.

DESCRIPTION OF USE

When the user first sets up Rasa in the command center, he unrolls his map and attaches it to a SmartBoard or other touch-sensitive surface (see Figure 5). A paper map, or in fact any Cartesian portrayal of the real world (e.g. photograph, drawing, etc.), can be registered to a position in the world by tapping at two points on it and speaking the coordinates for each. Immediately, Rasa is capable of projecting information on the paper map, or some other display, from its digital data sources. For example, Rasa can project unit symbology, other map annotations, 3D models, answers to questions, tables, etc.

As a user receives a radio report identifying an enemy reconnaissance company, (1) he draws a symbol denoting the unit on a Post-it.[1] Simultaneously, he can choose to modify the object with speech. For instance, he draws the reconnaissance company unit symbol in Figure 3 and at the same time gives the unit the name, "*Advanced guard*," via speech. (2) The system performs recognition of both speech and gesture in parallel, producing multiple hypotheses. (3) These hypotheses are parsed into meaning representations, and (4) are submitted for integration. (5) Some time later, the user places the Post-it on a registered map of the terrain at position 96-94. (6-8) This gesture is recognized and parsed, then also submitted for integration. After successful fusion of these inputs, Rasa says, "*Confirm: Enemy reconnaissance company called 'advanced guard' has been sighted at nine-six, nine-four.*" The user can then disconfirm the system's response if it is in error. If it is correct, the user need not confirm. Further action implies confirmation [17]. After confirmation, the unit is inserted into a database, which triggers a message to external digital systems. This example is illustrated in the diagram in Figure 6.

The next section describes the system architecture that makes this type of augmented, multimodal interaction possible. Following this description, we will examine this example of Rasa's use in further detail.

ARCHITECTURE

Rasa consists of autonomous and distributed software components that communicate using an agent communication language in the Adaptive Agent Architecture (AAA) [14], which is backwards compatible with the Open Agent Architecture (OAA) [6].

Agent Framework

The AAA is a robust, facilitated multi-agent system architecture specifically adapted for use with multimodal systems. A multi-platform Java agent shell provides services that allow each agent to interact with others in the agent architecture. The agents can dynamically join and leave the system. They register their capabilities with an AAA facilitator, which provides brokering and matchmaking services to them.

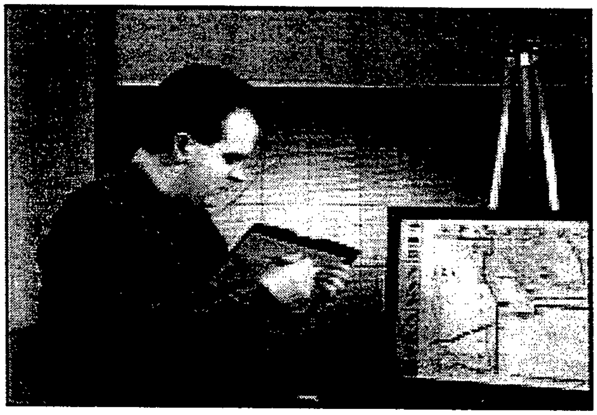

Figure 5. Photograph of Rasa.

Rasa's understanding of language is due to multiple recognition and understanding agents working in parallel and feeding their results to the multimodal integration agent. These agents and the human computer interface agents are described below.

User Interfaces

With Rasa, users first draw on a pad of Post-it notes affixed to a Cross Computing iPenPro™ digital pen tablet. A paper interface agent, running on the computer system connected to the tablet, captures digital ink, while the pen itself produces real ink on each note. However, there is no computer or user interface visible, other than the Post-its themselves.

In addition to the Post-its, map annotations can be created multimodally and then projected onto the paper map or a separate surface. As new units are placed on the map, a colored shadow indicating their position and their disposition (friendly or enemy) can be overlaid on the paper map. A table of all units present and their combat strength can be projected next to the map as visual confirmation of the status of each unit.[2] "Control measures," such as barbed wire, fortifications, berms, etc. can be drawn directly on the map's plastic overlay. The resulting military icon for that object is projected directly onto the map.

Conceptually, Rasa's user interfaces act as transparent interaction surfaces on the paper. Whenever the user touches the map and the touch-sensitive surface beneath it or uses the pen on the iPenPro tablet, they are interacting with Rasa. As they do, messages describing the digital ink left behind on those surfaces are sent to the facilitator for distribution to the relevant agents.

One additional invisible interface is used, text-to-speech, which can be enabled or disabled on command as needed. Each interaction with Rasa produces not only

[1] For simplicity, all units will be rectangular shaped.

[2] This type of table is often found next to the map tool in command posts.

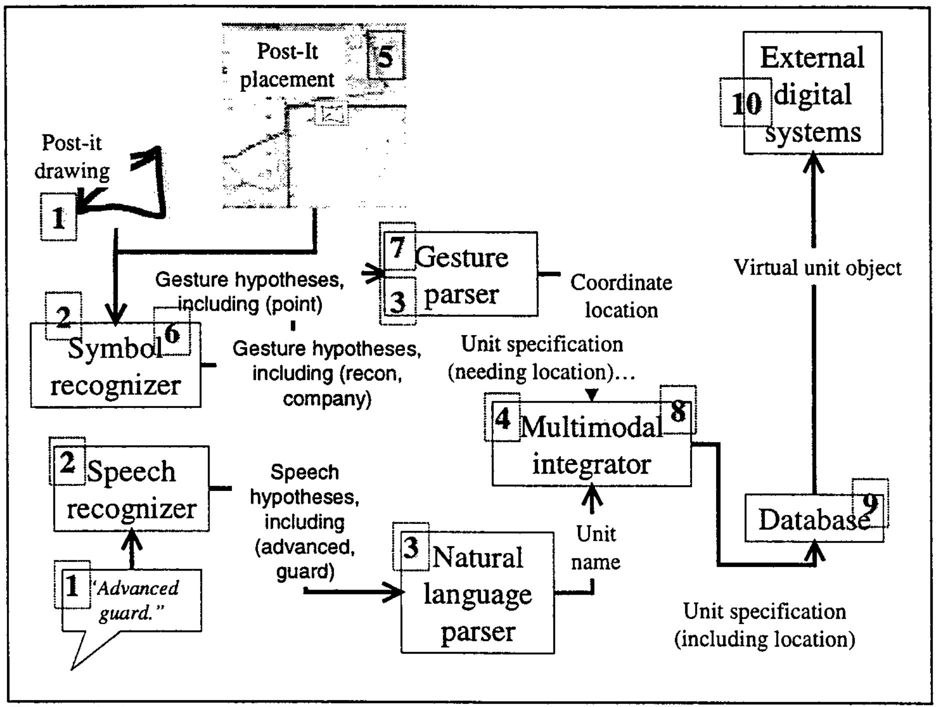
Post-It placement
5
Post-it drawing
1
Gesture hypotheses, including (point)
7 Gesture
3 parser
Coordinate location
External
10 digital
systems
Virtual unit object
2
Symbol 6
recognizer
Gesture hypotheses, including (recon, company)
Unit specification (needing location)…
4 Multimodal 8
integrator
2 Speech
recognizer
Speech hypotheses, including (advanced, guard)
3 Natural language parser
Unit name
Database 9
1 'Advanced guard."
Unit specification (including location)

Figure 6. An example of data and process flow in Rasa's system architecture

visual confirmation if a projector is present, but also audible confirmation.

Recognizers

Interaction with any of these paper surfaces results in ink being processed by Rasa's handwriting and gesture agents. Each produces simultaneous output to be processed by the multimodal integrator and combined as appropriate. At the same time, speech recognition is also enabled, providing input to the integrator. These agents and their abilities are discussed below.

Handwriting

Paragraph's writer-independent Calligrapher handwriting recognition engine has been incorporated as an agent into Rasa. Like the gesture agents described below, the handwriting agent receives input from interactions on the paper surface in the form of digital ink. The ink is sent from the user interfaces as individual strokes of time-stamped, contextualized, x-y pairs with supplementary information. The context supplied depends upon the user interface—maps provide a context of a location on the earth, while blank paper has no context.

Calligrapher can recognize natural letter shapes, including cursive, printed, and mixed case. Furthermore, given a vocabulary from the domain, it can distinguish between vocabulary, non-vocabulary, and non-handwriting (other ink-drawn gestures). Combining this ability with Rasa's other pen-based recognizers, Rasa can recognize and understand mixed symbolic and handwritten drawings, like that shown in Figure 6.

Gesture

Rasa's gesture agents recognize symbolic and editing gestures, such as points, lines, arrows, deletion, and grouping, as well as military symbology, including unit symbols and various control measures (barbed wire, fortification, boundaries, axes of advance etc.) based on a hierarchical recognition technique called Member-Team-Committee (MTC) [27].

The MTC weights the contributions of individual recognizers based on their empirically derived relative reliabilities, and thereby optimizes pattern recognition robustness. It uses a divide-and-conquer strategy, wherein the members produce local posterior estimates that are reported to one or more "team" leaders. The team leaders apply weighting to the scores, and pass results to the committee, which weights the distribution of the team's results. Using the MTC, the symbology recognizer can identify 200 different military unit symbols, while achieving a better than 90% recognition rate.

Speech

Rasa receives its spoken input from a microphone array attached to the top of the SmartBoard, directly above the map or from wireless microphones. The speech agent uses SAPI speech recognition engines from both Dragon Systems and Microsoft, or IBM's Voice Type Application Factory (VTAF), all continuous, speaker-independent recognizers. Training can be used to increase the accuracy of both SAPI-compliant engines. The SAPI engines use context-free grammars, while the VTAF engine uses bigrams. VTAF is limited to one hypothesis per spoken phrase; the two other engines produce n-best lists for each phrase (word scores are also available). Rasa's vocabulary is approximately 675 words, and the grammar specifies a far greater number of valid phrases.

The speech agent activates the recognizer when the user interacts with paper (touch to talk) or at all times (open microphone). The speech agent and microphone need not reside on any one computer in the augmented environment due to the distributed nature of the software architecture. Speech hypotheses, once generated, are forward to the natural language parser with their phrase scores and time stamps.

Parsers—producing understanding from recognition

Natural language

A definite-clause grammar produces typed feature structures—directed acyclic graphs (DAGs) of attribute value pairs, described more fully below—as meaning representation. For this task, the language consists of map-registration predicates, noun phrases that label entities, adverbial and prepositional phrases that supply additional information about the entity, and a variety of imperative constructs for supplying behavior to those entities or to control various systems.

Gesture

The gesture parser also produces typed feature structures, based on the list of recognition hypotheses and probability estimates supplied by the gesture recognizer. Typically, there would be multiple interpretations for each hypothesis. For example, a pointing gesture has at least three meaningful interpretations—a selection is being made, a location is being specified, or the first of many point locations are being specified. In the next section, we will examine how these multiple interpretations are weighed in the multimodal integrator.

Requirements for Multimodal Integration

Rasa's multimodal integration technology uses declarative rules to describe how the meanings of input from speech, gesture, or other modalities must be semantically and temporally compatible in order to combine. This fusion architecture was preceded by the original "Put-That-There" [2], and other approaches [5, 13, 20, 21]. However, as we reported in [11] these prior approaches are limited in four ways.

1. They are generally restricted to simple deictic gestural expressions.
2. They are primarily driven by the spoken modality; whereas first-class language exists in other modalities as well.
3. They have not provided a well-understood, generally applicable common meaning representation.
4. They have not provided a formally well-defined declarative mechanism for multimodal integration.

Our approach to overcoming each of these limitations supports:

- Multiple parallel recognizers and "understanders" that produce meaning fragments from continuous, parallel coordinated input streams.
- A common meaning representation—*typed feature structures*.
- A general application of *rule-based constraints* that satisfy, among other things, an empirically based [23], time-sensitive grouping process.
- A well-understood and semantically well-defined fusion algorithm that uses declarative rules for combining compatible meaning fragments—*unification*. Unification combines both complementary and redundant information, but rules out incompatible attribute values.
- A set of declarative multimodal grammar rules that enable parsing and interpretation of natural human input distributed across multiple simultaneous spatial dimensions, time, and speech.
- An algorithm that chooses the best semantically complete, joint interpretation of multimodal input, thus allowing one mode to compensate for another mode's errors [22].

In general, multimodal inputs are recognized, and then parsed, producing meaning descriptions in the form of typed feature structures. The integrator fuses these meanings together by evaluating any available integration rules for the type of input received and those partial inputs waiting in an integration buffer. Compatible types are unified, and the candidate meaning combination is subject to constraints. Successful unification and constraint satisfaction results in a new set of merged feature structures. The highest ranked semantically complete feature structure is executed. If none are complete, they wait in the buffer for further fusion, or contribute to the ongoing discourse as discussed below.

Typed Feature Structure Unification

Semantic compatibility is captured via unification over typed feature structures [3, 4]. Unification determines the consistency of two representational structures, and if consistent, combines them into a single feature structure. This type of unification is a generalization of term unification in logic programming languages, such as Prolog. However, feature structure unification differs from term unification because features are unordered attribute-value pairs of atoms and variables in a feature structure, rather than positionally encoded attributes in a term.

When two features structures are unified, a composite containing all of the feature specifications from each component structure is formed. Any feature common to both feature structures must have a compatible value. If the values of a common feature are atoms, they must be identical. If one is a variable, it becomes bound to the value of the corresponding feature in the other feature structure. If both are variables, they become constrained to always receive the same value. If the values are themselves feature structures, the unification operation is applied recursively. Importantly, feature structure unification results in a DAG structure when more than one value uses the same variable. Whatever value is ultimately unified with that variable will fill the value slot of all the corresponding features, resulting in a DAG.

Typed feature structures are an extension of the representation, whereby feature structures are assigned to hierarchically ordered types. Typed feature structure unification requires pairs of feature structures to be compatible in type (i.e., one must be in the transitive closure of the subtype relation with respect to the other). The result of a typed unification is the more specific feature structure in the type hierarchy. Typed feature structure unification is ideal for multimodal integration because it can combine complementary or redundant input from different modes, but rules out contradictory inputs.

In the next section, we describe in detail how Rasa's multimodal fusion works, which is described fully in [10, 11]. However, in the following section we will present a short example of its use in Rasa. We will also examine one of the examples of how Rasa supports human-computer discourse. We then briefly demonstrate the advantage of agent architectures in the support of human collaboration systems.

Multimodal Fusion in Rasa

To demonstrate how multimodal fusion works in Rasa, let's return to the example given above, in which an officer adds a new unit to Rasa's augmented map. Rasa responds in the following manner.

Speaking the unit's name ("advanced guard") generates a typed feature structure similar to that shown in Figure 7. To name a new unit, the name should be uttered while drawing a symbol that specifies the remaining constituents for a unit, such as the reconnaissance company symbol shown in Figure 3. This symbol is recognized as a reconnaissance company and assigned the feature structure in Figure 8.

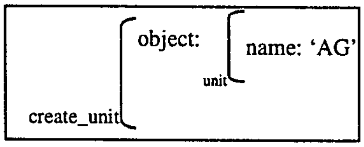

Figure 7. Typed feature structure from spoken utterance 'Advanced guard.'

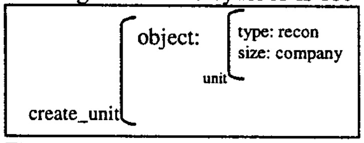

Figure 8. Typed feature structure resulting drawing recon company.

Rasa's fusion approach uses a multidimensional chart parser, or *multiparser*, based on chart parsing techniques from natural language processing [12]. Edges in the chart (feature structures of the multimodal input) are processed by multimodal grammar rules. Unification is able to ensure that the inputs contain compatible feature structures (attribute/values, where values can again be typed feature structures). A declarative set of temporal constraints is used that were developed based on empirical investigation of multimodal synchronization [23]. Spatial constraints are used for combining gestural inputs, and new constraints can be declared and applied in any rule.

In general, multimodal grammar rules are productions LHS → DTR1 DTR2; two daughter features (DTR1 and DTR2) are fused, under the constraints given, into the left-hand side.[3] The shared variables in the rules, denoted by numbers in square brackets, must unify appropriately with the inputs from the various modalities, as previously described.

One of Rasa's multimodal grammar rule, shown in Figure 9, declares that partially specified units (`dtr1` and `dtr2`) can combine with other partially specified units, so long as they are compatible in type, size, location, and name features, and they meet the constraints. It is expected that this rule will fire successfully when the user is attempting to create the particular unit using different modalities synchronously. `dtr2` is a placeholder for gestural input (note the location specification) and `dtr1` for spoken input, but this need not be the case. Figure 10 demonstrates partial application of the rule.

Any constraints are then satisfied using a Prolog meta-interpreter. For example, the timing constraints for this rule (the "overlap or follow" rule specification) guarantee that the two inputs will temporally overlap or that the

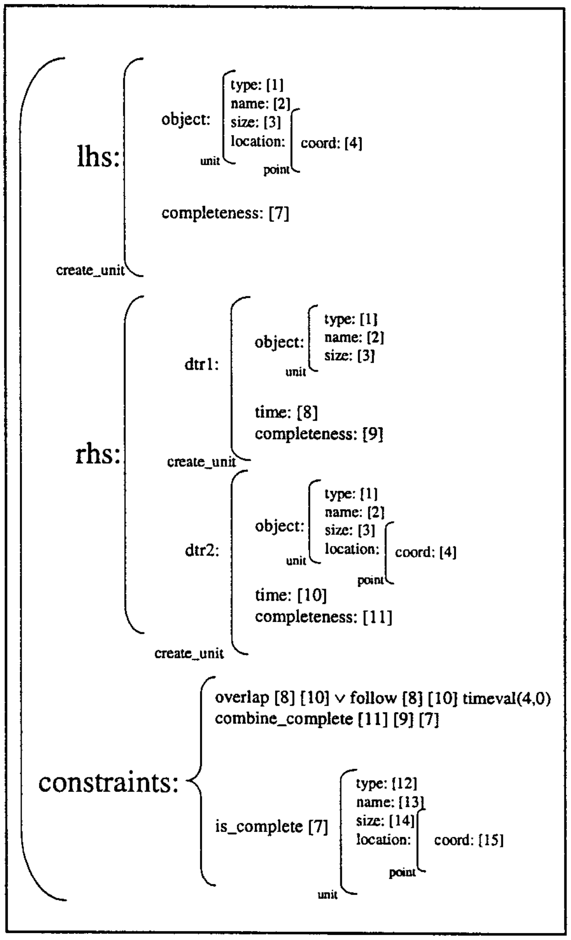

Figure 9. Multimodal grammar rule for partial unit fusion.

[3] In practice, any number of daughter features can be on the right-hand side of a rule.

gesture will precede the other input by at most 4 seconds.

Figure 10 shows that after fusion the left-hand-side is still missing a location feature for the unit specification. In the next section, we will describe how Rasa uses completeness criteria to notice this missing feature and query the user for it.

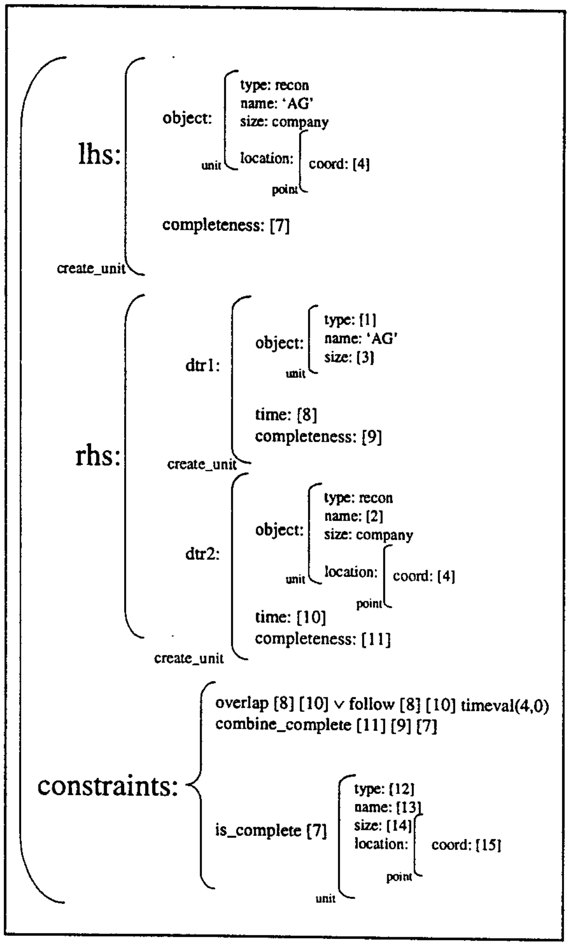

Figure 10. Application of grammar rule.

Discourse in Rasa

The completeness criterion and constraint work together to allow each feature structure to carry along a rule that specifies what features are needed to complete the structure. In this way, during feature structure evaluation, rules can fire that effectively instruct Rasa to formulate subdialogues with the user, to request the missing information.

For example, Figure 11 shows the completeness criteria rule for units. This feature structure captures the attribute names for each of the attributes that must have a value before the feature structure is found to be complete. In the example shown here, the criteria stipulates that every unit must have an object feature with values for type and size, as well as a location feature with feature structure type point and a coord value.

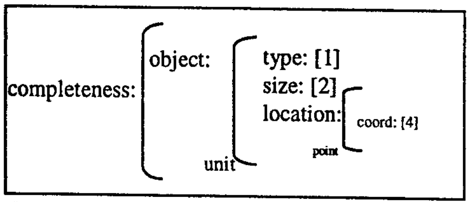

Figure 11. Completeness criterion for units.

This information can then be used by Rasa to produce queries when one of the values is missing. Rasa asks the user for the positions of units, after a tunable delay, that are fully specified except for the location feature. Users respond by placing the Post-it note on the map or by disconfirming the operation and throwing the unit away. If the Post-it note representing the reconnaissance company has not been placed on the map within 10 seconds, Rasa would respond by saying "Where is the reconnaissance company called 'Advanced guard'?"

Discourse rules can be declaratively specified in Rasa to promote complete mixed initiative collaborative dialogue. This is left for future work. However, human-computer dialogue is not the only aspect of collaboration supported by Rasa. Collaboration amongst multiple human users is also supported.

Collaboration with Rasa

Because user interfaces subscribe to and produce common messages, when they connect to the same facilitator, they immediately become part of a collaborative session. For instance, by subscribing to the entity-location database messages, a digital QuickSet user interface can be notified of changes in the locations of entities when a unit is moved on Rasa's paper map. Users can also couple their interfaces, to obtain tighter synchronicity. Coupled interfaces subscribe and produce common "ink" messages; meaning one user's ink appears on the others' map, immediately providing a shared drawing system.

DISCUSSION

Experimental evaluation of Rasa is underway. We are conducting a field test with military personnel to evaluate its performance against both paper map and computer-based map systems. These experiments should validate or disprove the arguments for opportunistic computing that we have made here.

Rasa and QuickSet's multimodal fusion architecture based on declarative multimodal integration rules makes it possible to easily declare the rules for Rasa's multi-interface input. The ability to declaratively specify these rules reduced the construction time of Rasa significantly. Having an agent framework and several useful agents available from the QuickSet system also fostered rapid assembly of the first Rasa prototype.

Opportunistic computing environments are not the only type of augmented environments to benefit from the adaptation of multimodal language. With colleagues from Columbia University [8], we are currently investigating how multimodal processing can be applied to systems that augment human senses. Specifically, we are adapting our multimodal fusion capability to support pointing and gesturing in virtual and augmented worlds.

RELATED WORK IN PAPER-BASED INTERFACES

Several recent approaches have successfully augmented paper in novel ways [1, 16, 19, 26]. However, none of these approaches treat the existing language of work as anything other than an annotation. These systems can capture the augmentations, but cannot understand them. Consequently, though they augment paper, and even support tasks involving written language, they are unable to take advantage of its properties as a means of augmenting the physical objects, as Rasa has done.

CONCLUDING REMARKS

We have presented Rasa, an environment for *opportunistic computing*, where physical objects are computationally augmented by a system observing users' work.

We have shown how language has properties that are especially suited for opportunistic computing environments. By virtue of the generativity of language, users can create augmentations; by virtue of its compositionality a large set of augmentations are possible; by virtue of its comprehensibility users other than the author can understand it; and by virtue of its persistence, the augmentation remains understandable in the face of failure.

Rasa leverages these benefits of language, by understanding the augmentations that officers in command posts place on Post-it notes and paper maps, resulting in the coupling of these placeholders with their digital counterparts.

We choose this approach because we feel we have no choice. The users have set aside their computational aids and have resorted to the paper tools. Our view is that users should not have to choose, but that we should begin to augment their choice of tools in a way that leaves them unchanged.

ACKNOWLEDGEMENTS

This work was supported in part by the Command Post of the Future Program, DARPA under contract number N66001-99-D-8503, and also in part by ONR grants: N00014-95-1-1164, N00014-99-1-0377, and N00014-99-1-0380. The Adaptive Agent Architecture infrastructure was supported by the DARPA CoABS program and US Air Force Research Laboratory under contract number: F30602-98-2-0098. Thanks to our colleagues, Richard Wesson and Sanjeev Kumar, and everyone who contributed to this work. The opinions and any errors that appear herein are the authors.

REFERENCES

1. Arai, T., Aust, D., and Hudson, S.E. PaperLink: A technique for hyperlinking from real paper to electronic content, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97),* (March 1997), ACM Press, 327-334.
2. Bolt, R.A., "Put-That-There": Voice and gesture at the graphics interface. *Computer Graphics,* 14(3 1980): 262-270.
3. Carpenter, R. Typed feature structures: Inheritance, (in)equality, and extensionality, in the *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing,,* (1990), Institute for Language Technology and Artificial Intelligence, Tilburg University, Tilburg, 9-18.
4. Carpenter, R., *The logic of typed feature structures.* Cambridge England: Cambridge University Press, 1992.
5. Cohen, P.R. Integrated interfaces for decision support with simulation, in the *Proceedings of the ACM Winter Simulation Conference,* (1991), ACM Press, 1066-1072.
6. Cohen, P.R., Cheyer, A., Wang, M., and Baeg, S.C. An open agent architecture, in the *Proceedings of the AAAI Spring Symposium,* (Madison, Wisconsin, March 1994), Reprinted in *Readings in Agents,* Huhns, M. and Singh, M. (eds.), Morgan Kaufmann Publishers, San Francisco, 1-8.
7. Cohen, P.R., Johnston, M., McGee, D.R., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: multimodal interaction for distributed applications, in the *Proceedings of the Fifth Annual ACM International Multimedia Conference (Multimedia '97),* (Seattle, WA, November 1997), ACM Press, 31-40.
8. Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. A Touring Machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment, in the *Proceedings of the International Symposium on Wearable Computing (ISWC'97),* (Cambridge, MA, October 13-14 1997, 74-81.
9. Ishii, H. and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97),* (Atlanta, GA, March 1997), ACM Press, 234-241.
10. Johnston, M. Unification-based multimodal parsing, in the *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 98),* (Montreal, Canada, August 1998), ACL Press, 624-630.

11. Johnston, M., Cohen, P.R., McGee, D.R., Oviatt, S.L., Pittman, J.A., and Smith, I. Unification-based multimodal integration, in the *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL '97),* (Madrid, Spain, March 1997), ACL Press, 281-288.

12. Kay, M. *Algorithm schemata and data structures in syntactic processing*, in *Readings in Natural Language Processing*, B.J. Grosz, K.S. Jones, and B.L. Webber, Editors. Morgan Kaufmann, 1986, 35-70.

13. Koons, D.B., Sparrell, C.J., and Thorisson, K.R. *Integrating simultaneous input from speech, gaze, and hand gestures*, in *Intelligent Multimedia Interfaces*, M.T. Maybury, Editor. AAAI Press/MIT Press: Cambridge, MA, 1993, 257-276.

14. Kumar, S., Cohen, P.R., and Levesque, H.J. The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams, in the *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000),* (Boston, MA, July 7-12 2000).

15. Leibe, B., Starner, T., Ribarsky, W., Wartell, Z., Krum, D., Singletary, B., and Hodges, L. The Perceptive Workbench: Toward spontaneous and natural interaction in semi-immersive virtual environments, in the *Proceedings of the IEEE Virtual Reality Conference (VR'2000),* (New Brunswick, NJ, Mar. 18-22 1999).

16. Mackay, W.E., Fayard, A.L., Frobert, L., and Médini, L. Reinventing the familiar: Exploring an augmented reality design space for air traffic control, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'98),* (Los Angeles, CA, April 18-23 1998), ACM Press, 558-565.

17. McGee, D.R., Cohen, P.R., and Oviatt, S. Confirmation in multimodal systems, in the *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics (COLING-ACL '98),* (Montreal, Quebec, Canada, August 1998), ACL Press, 823-829.

18. McGee, D.R., Cohen, P.R., and Wu, L. Something from nothing: Augmenting a paper-based work practice with multimodal interaction, in the *Proceedings of the ACM Conference on Designing Augmented Reality Environments (DARE'00),* (Helsingor, Denmark, April 12-14 2000), ACM Press, 71-80.

19. Moran, T.P., Saund, E., Melle, W.v., Bryll, R., Gujar, A.U., Fishkin, K.P., and Harrison, B.L. The ins and outs of collaborative walls: Demonstrating the Collaborage concept, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99),* (Pittsburgh, PA, May 15-20 1999), ACM Press, CHI'99 Extended Abstracts, 192-193.

20. Neal, J.G. and Shapiro, S.C. *Intelligent multi-media interface technology*, in *Intelligent User Interfaces*, J.W. Sullivan and S.W. Tyler, Editors. ACM Press, Frontier Series, Addison Wesley Publishing Co.: New York, New York, 1991, 45-68.

21. Nigay, L. and Coutaz, J. A generic platform for addressing the multimodal challenge, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95),* (Denver, CO, May 7-11 1995), ACM Press, 98-105.

22. Oviatt, S.L. Mutual disambiguation of recognition errors in a multimodal architecture, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'99),* ACM Press, 576-583.

23. Oviatt, S.L., DeAngeli, A., and Kuhn, K. Integration and synchronization of input modes during multimodal human-computer interaction, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97),* (Atlanta, GA, March 1997), ACM Press, 415-422.

24. Streitz, N.A., Geibler, J., and Holmer, T. Roomware for cooperative buildings: integrated design of architectural spaces and information spaces, in the *Proceedings of the First International Workshop on Cooperative Buildings: Integrating Information, Organization, and Architecture (CoBuild '98),* (Darmstadt, Germany, February 1998), Springer-Verlag, 4-21.

25. Underkoffler, J. and Ishii, H. Urp: a luminous-tangible workbench for urban planning and design, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99),* (Pittsburgh, PA, May 1999), ACM Press, 386-393.

26. Want, R., Fishkin, K.P., Gujar, A., and Harrison, B.L. Bridging physical and virtual worlds with electronic tags, in the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99),* (Pittsburgh, PA, May 1999), ACM Press, 370-377.

27. Wu, L., Oviatt, S., and Cohen, P., Multimodal integration - A statistical view. *IEEE Transactions on Multimedia*, 1(4), 1999: 334-341.